

What is claimed is:

CLAIMS

- 1 1. A computer system that employs a plurality of execution threads to perform tasks
2 that the threads identify dynamically, the computer system being so programmed as to:
 - 3 A) provide a plurality of task queues, each of which is associated with a dif-
4 ferent ordered pair of the threads, one thread of the ordered pair being de-
5 nominated the enqueuer of that queue and the other being denominated the
6 dequeuer thereof;
 - 7 B) when one said thread identifies a task, pushes an identifier of that the task
8 thus identified onto a set of at least one of the queues of which that thread
9 is an enqueuer; and
 - 10 C) when one said thread requires one of the dynamically identified tasks to
11 perform, causes that thread to perform a task identified by a task identifier
12 fetched by that thread from a task queue of which that thread is the de-
13 queuer.
- 1 2. A computer system as defined in claim 1 wherein each said dynamically identi-
2 fied task is the garbage-collection task of performing, for a given object associated with
3 that task, processing that includes identifying in the given object references to other ob-
4 jects and thereby identifying the tasks of performing similar processing for those other
5 objects.
- 1 3. A computer system as defined in claim 1 wherein each said task identifier is an
2 identifier of the object with which the task is associated.
- 1 4. A computer system as defined in claim 3 wherein each said dynamically identi-
2 fied task is the garbage-collection task of performing, for a given object associated with
3 that task, processing that includes identifying in the given object references to other ob-
4 jects and thereby identifying the tasks of performing similar processing for those other
5 objects.

1 5. A computer system as defined in claim 3 wherein each said task identifier is a
2 pointer to the object with which the task is associated

1 6. A computer system as defined in claim 1 wherein, when one said thread identifies
2 a task, the computer system pushes an identifier of that thread onto only one of the
3 queues of which that thread is an enqueueer.

1 7. A computer system as defined in claim 6 wherein each said dynamically identi-
2 fied task is the garbage-collection task of performing, for a given object associated with
3 that task, processing that includes identifying in the given object references to other ob-
4 jects and thereby identifying the tasks of performing similar processing for those other
5 objects.

1 8. A computer system as defined in claim 1 wherein identifiers of tasks successively
2 identified by a given thread are not in general pushed onto the same queue.

1 9. A computer system as defined in claim 8 wherein each said dynamically identi-
2 fied task is the garbage-collection task of performing, for a given object associated with
3 that task, processing that includes identifying in the given object references to other ob-
4 jects and thereby identifying the tasks of performing similar processing for those other
5 objects.

1 10. A computer system as defined in claim 1 wherein a task queue is provided for
2 each ordered pair of the threads.

1 11. A computer system as defined in claim 10 wherein each said dynamically identi-
2 fied task is the garbage-collection task of performing, for a given object associated with
3 that task, processing that includes identifying in the given object references to other ob-

jects and thereby identifying the tasks of performing similar processing for those other objects.

12. For using a computer system to employ a plurality of execution threads to perform tasks that the threads identify dynamically, a method that includes:

- A) providing a plurality of task queues, each of which is associated with a different ordered pair of the threads, one thread of the ordered pair being denominated the enqueueer of that queue and the other being denominated the dequeuer thereof;
- B) when one said thread identifies a task, pushing an identifier of that the task thus identified onto a set of at least one of the queues of which that thread is an enqueueer; and
- C) when one said thread requires one of the dynamically identified tasks to perform, causing that thread to perform a task identified by a task identifier fetched by that thread from a task queue of which that thread is the dequeuer.

13. A method as defined in claim 12 wherein each said dynamically identified task is the garbage-collection task of performing, for a given object associated with that task, processing that includes identifying in the given object references to other objects and thereby identifying the tasks of performing similar processing for those other objects.

14. A method as defined in claim 12 wherein each said task identifier is an identifier of the object with which the task is associated.

15. A method as defined in claim 14 wherein each said dynamically identified task is the garbage-collection task of performing, for a given object associated with that task, processing that includes identifying in the given object references to other objects and thereby identifying the tasks of performing similar processing for those other objects.

1 16. A method as defined in claim 14 wherein each said task identifier is a pointer to
2 the object with which the task is associated

1 17. A method as defined in claim 12 wherein, when one said thread identifies a task,
2 the computer system pushes an identifier of that thread onto only one of the queues of
3 which that thread is an enqueueer.

1 18. A method as defined in claim 17 wherein each said dynamically identified task is
2 the garbage-collection task of performing, for a given object associated with that task,
3 processing that includes identifying in the given object references to other objects and
4 thereby identifying the tasks of performing similar processing for those other objects.

1 19. A method as defined in claim 12 wherein identifiers of tasks successively identi-
2 fied by a given thread are not in general pushed onto the same queue.

1 20. A method as defined in claim 19 wherein each said dynamically identified task is
2 the garbage-collection task of performing, for a given object associated with that task,
3 processing that includes identifying in the given object references to other objects and
4 thereby identifying the tasks of performing similar processing for those other objects.

1 21. A method as defined in claim 12 wherein a task queue is provided for each or-
2 dered pair of the threads.

1 22. A method as defined in claim 21 wherein each said dynamically identified task is
2 the garbage-collection task of performing, for a given object associated with that task,
3 processing that includes identifying in the given object references to other objects and
4 thereby identifying the tasks of performing similar processing for those other objects.

1 23. A storage medium containing instructions readable by a computer system to con-
2 figure the computer system to employ a plurality of execution threads to perform dy-
3 namically identified tasks by:

- 4 A) providing a plurality of task queues, each of which is associated with a dif-
5 ferent ordered pair of the threads, one thread of the ordered pair being de-
6 nominated the enqueueer of that queue and the other being denominated the
7 dequeuer thereof;
8 B) when one said thread identifies a task, pushing an identifier of that the task
9 thus identified onto a set of at least one of the queues of which that thread
10 is an enqueueer; and
11 C) when one said thread requires one of the dynamically identified tasks to
12 perform, causing that thread to perform a task identified by a task identi-
13 fier fetched by that thread from a task queue of which that thread is the
14 dequeuer.

1 24. A storage medium as defined in claim 23 wherein each said dynamically identi-
2 fied task is the garbage-collection task of performing, for a given object associated with
3 that task, processing that includes identifying in the given object references to other ob-
4 jects and thereby identifying the tasks of performing similar processing for those other
5 objects.

1 25. A storage medium as defined in claim 23 wherein each said task identifier is an
2 identifier of the object with which the task is associated.

1 26. A storage medium as defined in claim 25 wherein each said dynamically identi-
2 fied task is the garbage-collection task of performing, for a given object associated with
3 that task, processing that includes identifying in the given object references to other ob-
4 jects and thereby identifying the tasks of performing similar processing for those other
5 objects.

1 27. A storage medium as defined in claim 25 wherein each said task identifier is a
2 pointer to the object with which the task is associated

1 28. A storage medium as defined in claim 23 wherein, when one said thread identifies
2 a task, the computer system pushes an identifier of that thread onto only one of the
3 queues of which that thread is an enqueueer.

1 29. A storage medium as defined in claim 28 wherein each said dynamically identi-
2 fied task is the garbage-collection task of performing, for a given object associated with
3 that task, processing that includes identifying in the given object references to other ob-
4 jects and thereby identifying the tasks of performing similar processing for those other
5 objects.

1 30. A storage medium as defined in claim 23 wherein identifiers of tasks successively
2 identified by a given thread are not in general pushed onto the same queue.

1 31. A storage medium as defined in claim 30 wherein each said dynamically identi-
2 fied task is the garbage-collection task of performing, for a given object associated with
3 that task, processing that includes identifying in the given object references to other ob-
4 jects and thereby identifying the tasks of performing similar processing for those other
5 objects.

1 32. A storage medium as defined in claim 23 wherein a task queue is provided for
2 each ordered pair of the threads.

1 33. A storage medium as defined in claim 32 wherein each said dynamically identi-
2 fied task is the garbage-collection task of performing, for a given object associated with
3 that task, processing that includes identifying in the given object references to other ob-
4 jects and thereby identifying the tasks of performing similar processing for those other
5 objects.

1 34. A computer signal representing a sequence of instructions that, when executed by
2 a computer system, cause the computer system to employ a plurality of execution threads
3 to perform dynamically identified tasks by:

- 4 A) provide a plurality of task queues, each of which is associated with a dif-
5 ferent ordered pair of the threads, one thread of the ordered pair being de-
6 nominated the enqueuer of that queue and the other being denominated the
7 dequeuer thereof;
8 B) when one said thread identifies a task, pushes an identifier of that the task
9 thus identified onto a set of at least one of the queues of which that thread
10 is an enqueuer; and
11 C) when one said thread requires one of the dynamically identified tasks to
12 perform, causes that thread to perform a task identified by a task identifier
13 fetched by that thread from a task queue of which that thread is the de-
14 queuer.

1 35. A computer signal as defined in claim 34 wherein each said dynamically identi-
2 fied task is the garbage-collection task of performing, for a given object associated with
3 that task, processing that includes identifying in the given object references to other ob-
4 jects and thereby identifying the tasks of performing similar processing for those other
5 objects.

1 36. A computer signal as defined in claim 34 wherein each said task identifier is an
2 identifier of the object with which the task is associated.

1 37. A computer signal as defined in claim 36 wherein each said dynamically identi-
2 fied task is the garbage-collection task of performing, for a given object associated with
3 that task, processing that includes identifying in the given object references to other ob-
4 jects and thereby identifying the tasks of performing similar processing for those other
5 objects.

1 38. A computer signal as defined in claim 36 wherein each said task identifier is a
2 pointer to the object with which the task is associated

1 39. A computer signal as defined in claim 34 wherein, when one said thread identifies
2 a task, the computer system pushes an identifier of that thread onto only one of the
3 queues of which that thread is an enqueueer.

1 40. A computer signal as defined in claim 39 wherein each said dynamically identi-
2 fied task is the garbage-collection task of performing, for a given object associated with
3 that task, processing that includes identifying in the given object references to other ob-
4 jects and thereby identifying the tasks of performing similar processing for those other
5 objects.

1 41. A computer signal as defined in claim 34 wherein identifiers of tasks successively
2 identified by a given thread are not in general pushed onto the same queue.

1 42. A computer signal as defined in claim 41 wherein each said dynamically identi-
2 fied task is the garbage-collection task of performing, for a given object associated with
3 that task, processing that includes identifying in the given object references to other ob-
4 jects and thereby identifying the tasks of performing similar processing for those other
5 objects.

1 43. A computer signal as defined in claim 34 wherein a task queue is provided for
2 each ordered pair of the threads.

1 44. A computer signal as defined in claim 43 wherein each said dynamically identi-
2 fied task is the garbage-collection task of performing, for a given object associated with
3 that task, processing that includes identifying in the given object references to other ob-

jects and thereby identifying the tasks of performing similar processing for those other objects.

45. A computer system that employs a plurality of execution threads to perform tasks that the threads identify dynamically, the computer system including:

- A) means for providing a plurality of task queues, each of which is associated with a different ordered pair of the threads, one thread of the ordered pair being denominated the enqueueer of that queue and the other being denominated the dequeueer thereof;
- B) means for, when one said thread identifies a task, pushing an identifier of that the task thus identified onto a set of at least one of the queues of which that thread is an enqueueer; and
- C) means for, when one said thread requires one of the dynamically identified tasks to perform, causing that thread to perform a task identified by a task identifier fetched by that thread from a task queue of which that thread is the dequeueer.